

Univerzitet u Sarajevu

Elektrotehnički fakultet u Sarajevu

Računarstvo i informatika

eParking

Dokumentacija koda

Predmet: Objektno orijentisana analiza i dizajn

Članovi grupe: Milica Đokić, Faris Demir, Amar Bešlagić

Juni, 2020.

U ovom dokumentu ćemo pojasniti strukturu našeg projekta kao i pojedine detalje vezane za implementaciju funkcionalnosti koje nudi. Tema našeg projekta je eParking ASP.NET Core web aplikacija koja prati MVC patern.

Klikom na navigacijski link Mapa na vrhu ekrana otvorit će se Map view preko istoimene metode u EParking kontroleru. U Map view smo dodali i 3 button-a a to su: button za iscrtavanje rute na mapi, button za registraciju ulaska na parking i button za brisanje trenutno prikazane rute. Za prikaz mape koristimo JavaScript funkciju initMap gdje kreiramo objekat google.maps.Map i dodamo mu iz baze sve postojeće parkinge kako bi se kreirali odgovarajući markeri na mapi. Također u istoj funkciji kreiramo i info dijaloge koji se prikažu klikom na markere na mapi tako što dodamo listener na svaki marker. Što se tiče računanja rute, koristimo funkciju calculateAndShowRoute, te pomoću objekata DirectionsService i DirectionsRenderer te funkcije distanceBetween prikazujemo rutu na mapi od trenutne lokacije pa do najbližeg parkinga ili ako je član logovan do njegovog rezervisanog parkinga. Pomoću funkcije removeRoute brišemo iscrtanu rutu sa mape tako što postavimo rute na prazan niz te nakon toga onemogućimo klik na button za registraciju ulaska na parking.

Klikom na button za registraciju ulaska na parking otvara se Timer view preko istoimene metode u EParking kontroleru. Prije nego što se ovaj view otvori, preko post metode Map pošaljemo određenu parking lokaciju kako bi Timer view imao sve potrebne podatke. Na Timer view imamo tri kolone sa podacima. U sredini se nalazi timer koji prikazuje vrijeme provedeno na parkingu i on je realizovan pomoću JavaScript funkcije stopWatch koja svake sekunde ažurira brojeve na ekranu koristeći funkciju setInterval. Pomoću JavaScript funkcije startStop regulišemo zaustavljanje timera kada se klikne na button za registrovanje izlaska sa parkinga. Također prikazujemo i cijenu koju korisnik treba platiti preko JavaScript funkcije priceChange a koja se uživo ažurira u zavisnosti od cijene parkinga (cijena na osnovu koje se obračunava parking se dobije pomoću metode PrilagodiCijenu u Eparking kontroleru koja na osnovu doba dana vraća ili dnevnu ili noćnu satnu cijenu parkinga). Klikom na dugme za registraciju izlaska sa parkinga, pomoću post metode Timer šaljemo iznos koji treba prikazati korisniku na Pay view kako bi mogao izvršiti plaćanje istog. Pay view sadrži tab za plaćanje karticom, plaćanje putem PayPal-a i plaćanje gotovinom. Nakon uspješnog plaćanja korisniku se prikaže alert da je uspješno platio parking. Alert se kreira u Pay view u slučaju da smo inicijalizirali ViewBag.Alert u Pay post metodi na vrijednost „'show“.

Kada kliknemo na Login, unosimo username i password. U EParkingControlleru u metodi Login, se prolazi kroz sve tabele i provjerava se da li postoji korisnik sa tim username-om i password-om, nakon čega se otvara account tog korisnika.

U kontroleru ClanController u metodi Account, se provjerava datum registracije vozila i tip članarine, te ukoliko je članarina istekla status se postavlja na INACTIVE. Osim toga se uzmu podaci o odgovarajućem vozilu, i pošalju se u Account view za tog člana.

U kontroleru VlasnikController u metodi Account se pokupi lista zahtjeva koji su predati tom vlasniku, i pošalje se u Account view vlasnika, kako bi se mogla prikazati u accountu.

U kontroleru AdministratorController u metodi Account se pakuje informacije o ukupnom broju parkinga, vlasnika i članova sa aktivnom članarinom, i te informacije se šalju u Account view Administratora.

U svakom accountu postoji mogućnost mijenjanja informacija, pritiskom na dugme Uredi profil. Editovanje accounta se vrši kroz metodu Edit u kontrolerima. Član ima mogućnost da vidi informacije o sebi i o vozilu. Vlasnik može pregledati informacije o sebi, može pregledati analitiku prihoda, kao i listu zahtjeva. U ZahtjevControlleru u metodi OdobravanjeZahtjeva se status člana mijenja u aktivan, vlasniku se ažuriraju prihodi, i briše se zahtjev. Također, korisniku se šalje konfirmacijski email. U metodi OdbijanjeZahtjeva se briše zahtjev a status članarine ostaje neaktivan. Osim toga, vlasnik može promijeniti cijenu svog parkinga, klikom na button Promijeni cjenovnik. To se dešava i metodi Izmjena u VlasnikControlleru.

Kada administrator uđe na svoj profil, ima mogućnost prikaza Index View-a za članove, parking lokacije i vlasnike. Može brisati postojeće članove, što se obavlja u metodi Delete u ClanControlleru. Može dodavati nove (kroz metodu Create u VlasnikControlleru) i brisati postojeće vlasnike (kroz metodu Delete). Što se tiče parking lokacija, može dodavati (metoda Create u ParkingLokacijaController), mijenjati (metoda Edit u ParkingLokacijaController) i brisati (metoda Delete u ParkingLokacijaController).

Ukoliko izaberemo opciju Register, otvara se prozor u kojem unosimo prvo podatke o korisniku, zatim podatke o vozilu i na kraju se šalje zahtjev vlasniku parkinga. Dodavanje objekata u bazu se vrši kroz metodu Create u kontrolerima.

Klikom na navigacijski link About, prikazuju se informacije o projektu i o članovima tima.

Ukoliko je neki korisnik uradio Login, prikazuju se novi navigacijski linkovi Moj profil i Log out. Klikom na Moj profil korisnik se vraća na početni prozor svog profila, a klikom na Log out se vraća na prozor za Login.

Ukoliko bi se dalje nadograđivala aplikacija, bilo bi potrebno detaljnije uraditi procese plaćanja, tj. povezati plaćanje sa npr. sistemom banke i sistemom za PayPal, kako bi se na taj način radila validacija podataka vezanih za kreditnu karticu i PayPal račun.

Deployment aplikacije je urađen i kreiran je link na osnovu kojeg se može pristupiti aplikaciji: <https://eparkingood.azurewebsites.net>